



# PIO-821 Series Classic Driver DLL Software Manual

Version 1.0, Feb. 2014

## SUPPORTS

Board includes PIO-821L, PIO-821H, PIO-821LU and PIO-821HU.

## WARRANTY

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

## WARNING

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

## COPYRIGHT

Copyright © 2014 by ICP DAS. All rights are reserved.

## TRADEMARK

Names are used for identification only and may be registered trademarks of their respective companies.

## CONTACT US

If you have any question, please feel to contact us at:

[service@icpdas.com](mailto:service@icpdas.com); [service.icpdas@gmail.com](mailto:service.icpdas@gmail.com)

We will give you quick response within 2 workdays.



# **TABLE OF CONTENTS**

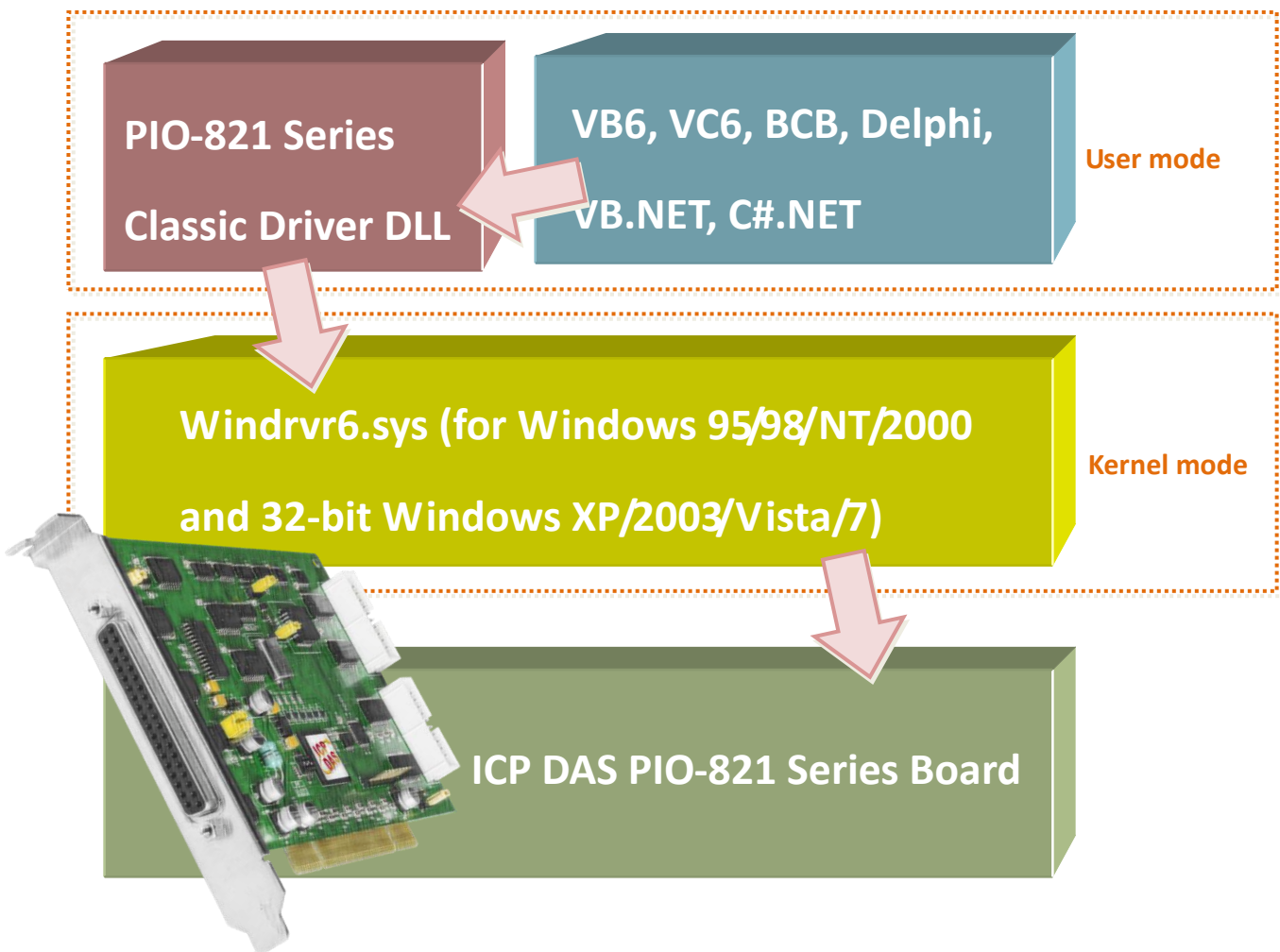
<b>1.</b>	<b>INTRODUCTION .....</b>	<b>3</b>
1.1	OBTAINING THE DRIVER INSTALLER PACKAGE .....	4
1.2	DRIVER INSTALLING PROCEDURE .....	5
1.3	PNP DRIVER INSTALLATION .....	8
1.4	UNINSTALLING THE PIO-821 SERIES CLASSIC DRIVER .....	10
<b>2.</b>	<b>DLL FUNCTION DESCRIPTIONS .....</b>	<b>11</b>
2.1	ERROR CODE TABLE .....	14
2.2	DRIVER FUNCTIONS.....	15
	<i>PIO821_GetDllVersion</i> .....	15
	<i>PIO821_ActiveBoard</i> .....	15
	<i>PIO821_CloseBoard</i> .....	16
	<i>PIO821_TotalBoard</i> .....	16
	<i>PIO821_GetCardInf</i> .....	17
	<i>PIO821_IsBoardActive</i> .....	18
2.3	ANALOG OUTPUT FUNCTIONS .....	19
	<i>PIO821_DA_Hex</i> .....	19
	<i>PIO821_DA</i> .....	20
2.4	EEPROM FUNCTIONS.....	21
	<i>PIO821_WriteEEP</i> .....	21
2.5	DIGITAL INPUT/OUTPUT FUNCTIONS .....	22
	<i>PIO821_DigitalIn</i> .....	22
	<i>PIO821_DigitalOut</i> .....	23
	<i>PIO821_InputByte</i> .....	24
	<i>PIO821_OutputByte</i> .....	25
	<i>PIO821_InputWord</i> .....	26
	<i>PIO821_OutputWord</i> .....	27
2.6	TIMER/COUNTER FUNCTIONS .....	28
	<i>PIO821_SetCounter</i> .....	28
	<i>PIO821_ReadCounter</i> .....	29
2.7	ANALOG INPUT FUNCTIONS .....	30
	<i>PIO821_SetChannelConfig</i> .....	30
	<i>PIO821_Delay</i> .....	31
	<i>PIO821_ADPollingHex</i> .....	32

<i>PIO821_ADPolling</i> .....	33
<i>PIO821_ADsPolling</i> .....	34
<i>PIO821_ADsPacer</i> .....	35
<b>2.8 INTERRUPT FUNCTIONS</b> .....	<b>36</b>
<i>PIO821_InstallIrq</i> .....	36
<i>PIO821_IntADStart</i> .....	36
<i>PIO821_GetADsfloat</i> .....	37
<i>PIO821_GetADsHex</i> .....	38
<i>PIO821_RemoveIrq</i> .....	38
<b>3. DEMO PROGRAMS</b> .....	<b>39</b>
3.1 FOR MICROSOFT WINDOWS .....	39
3.2 FOR DOS .....	43
3.2.1 <i>LIB (PIO.H) Function Description</i> .....	44

# 1. Introduction

The software is a collection of digital I/O, analog I/O and Timer/Counter subroutines for PIO-821 series card add-on cards for **Windows 95/98/NT/2000** and **32-bit Windows XP/2003/Vista/7** applications. The application structure is presented in the following diagram.

The subroutines in **PIO821.DLL** are easy understanding as its name standing for. It provides powerful, easy-to-use subroutine for developing your data acquisition application. Your program can call these DLL functions by **VB, VC, Delphi, BCB, VB.NET 2005** and **C#.NET 2005** easily. Then the DLL driver will bypass the function call to **Windrvr6.sys** in order to access the hardware system. To speed-up your developing process, some demonstration source program are provided.



## 1.1 Obtaining the Driver Installer Package

PIO-821 series card can be used on Linux and Windows 95/98/NT/2000 and 32-bit XP/2003/Vista/7 based systems, and the drivers are fully Plug and Play (PnP) compliant for easy installation.

The driver installer package for the PIO-821 series can be found on the supplied CD-ROM, or can be obtained from the ICP DAS FTP web site. The location and addresses are indicated in the table below:



**CD:\\ NAPDOS\\PCI\\PIO-821\\DLL\\**



**<ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-821/dll/>**



**<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-821/dll/driver/>**

## 1.2 Driver Installing Procedure

Before the driver installation, you must complete the hardware installation. For detailed information about the hardware installation, please refer to hardware user manual of PIO-821 series card.

The hardware user manual is contained in:

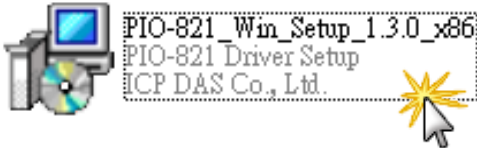


CD:\NAPDOS\PCI\PIO-821 \Manual\



<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-821/manual/>

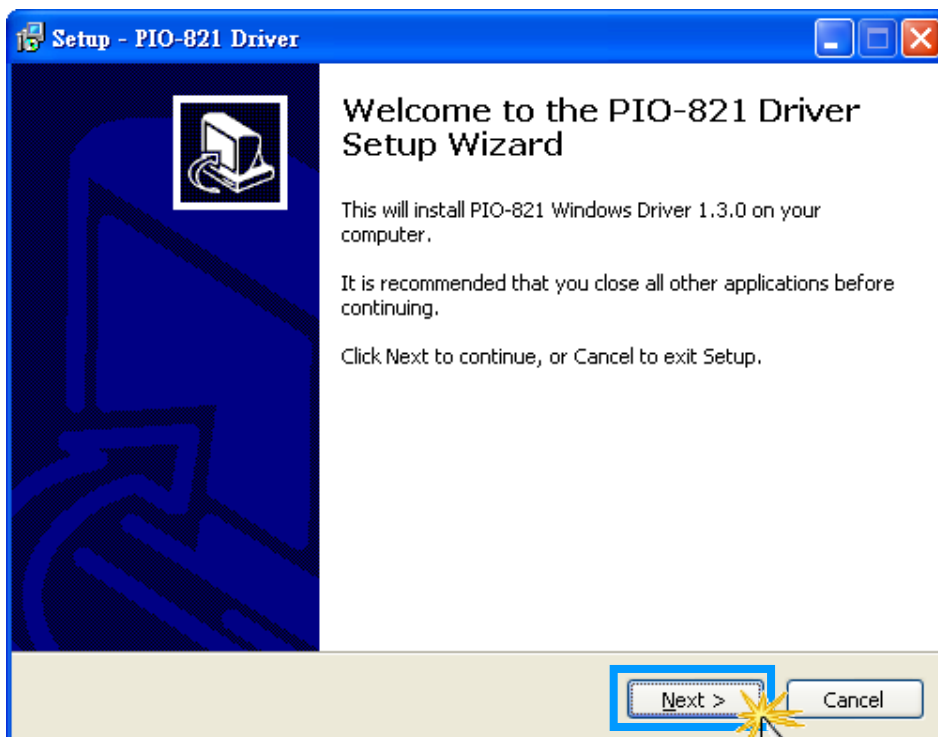
To install the PIO-821 series classic drivers, follow the procedure described below:



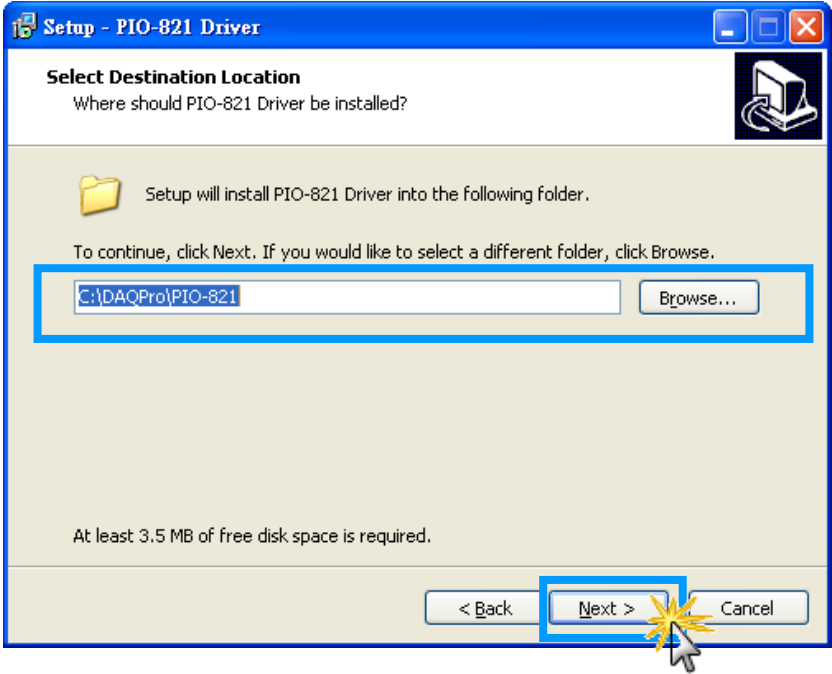
Step 1: Double-Click

“PIO-821\_Win\_Setup\_xxxx.exe” to install driver.

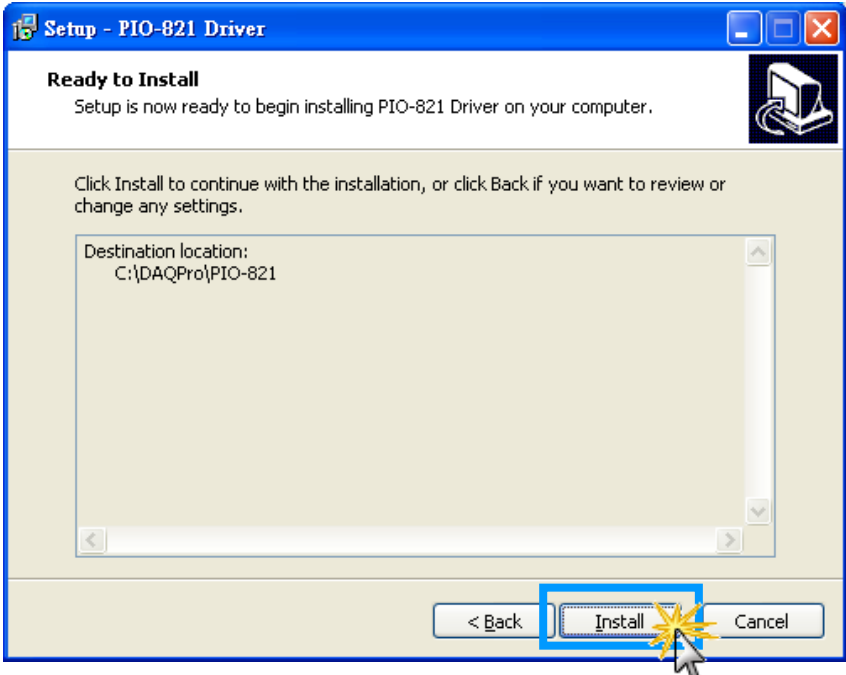
Step 2: Click the “**Next>**” button to start the installation on the “**Setup – PI-821 Driver**” window.



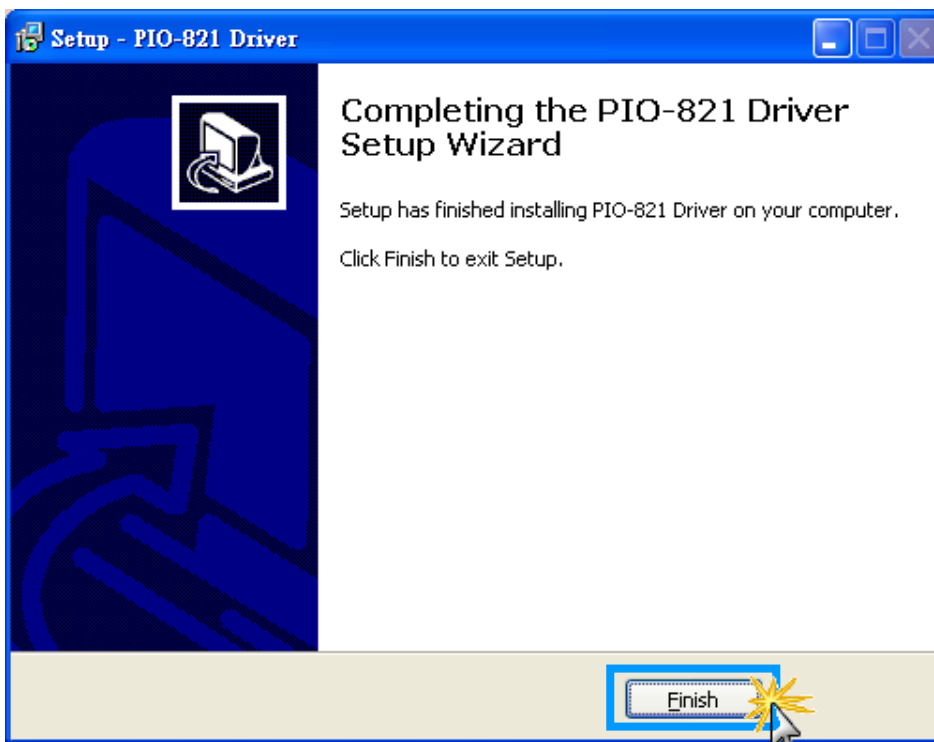
Step 3: Click the “**N**ext>” button to install the driver into the **default** folder.



Step 4: Click the “**I**nstall” button to continue the installation.



Step 5: Click the “**Finish**” button.





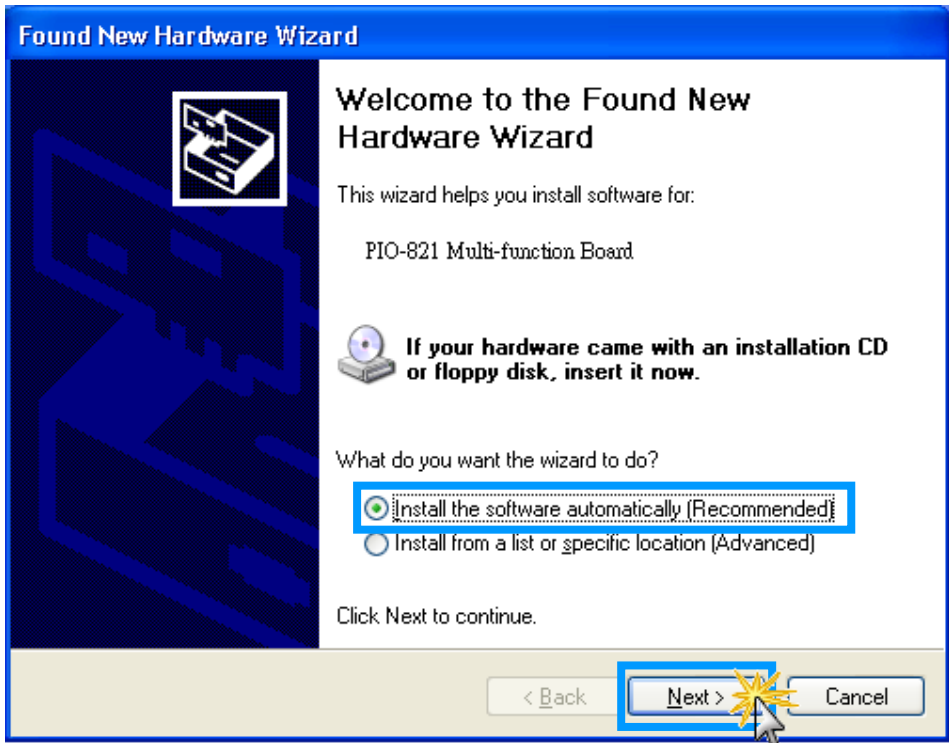
### 1.3 PnP Driver Installation

Step 1: The system should find the new card and then continue to finish the Plug&Play steps.

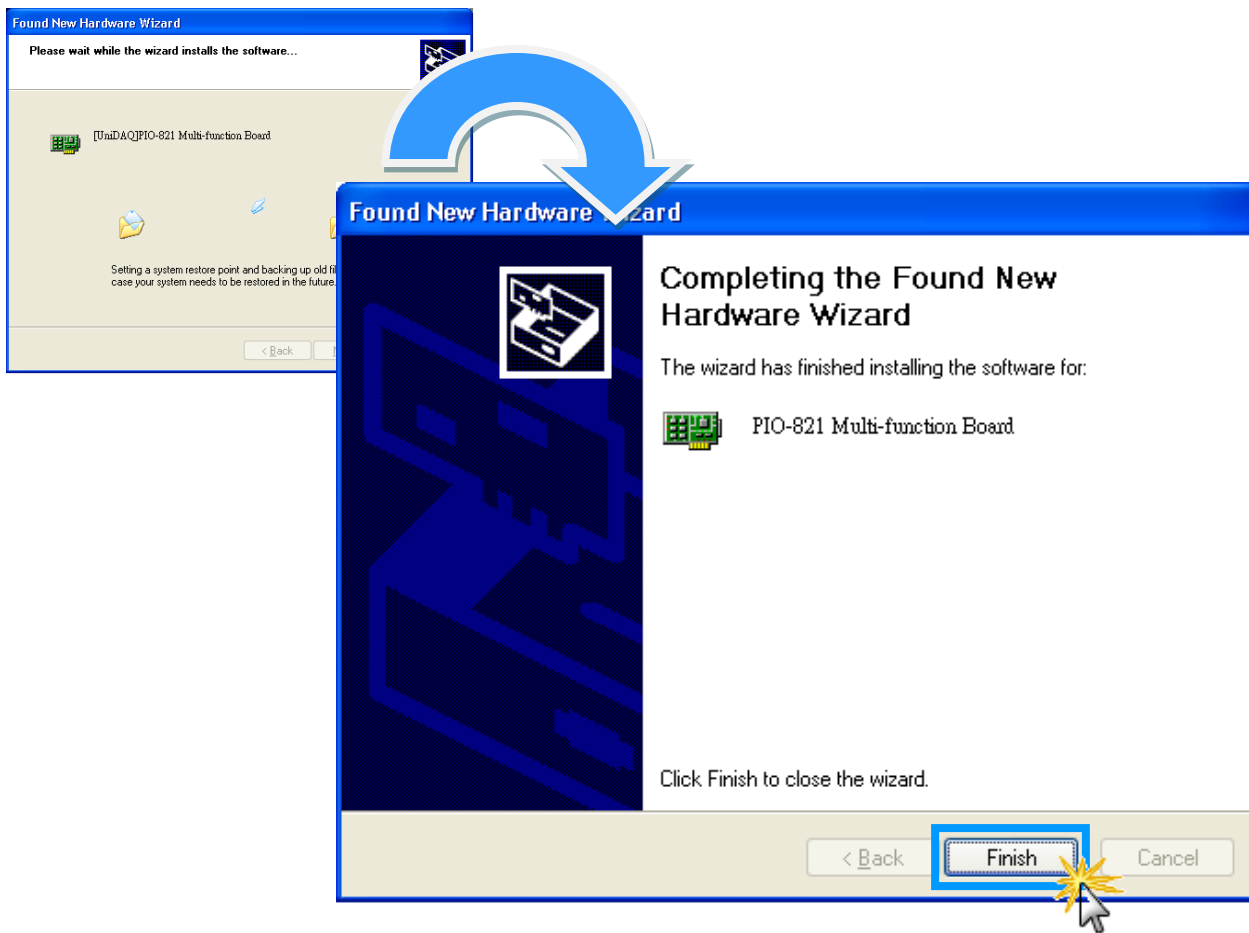
*Note: Some operating system (such as Windows Vista/7) will find the new card and make it work automatically, so the Step2 to Step4 will be skipped.*



Step 2: Select “Install the software automatically [Recommended]” and click the “Next>” button.



Step 3: Click the **“Finish”** button.



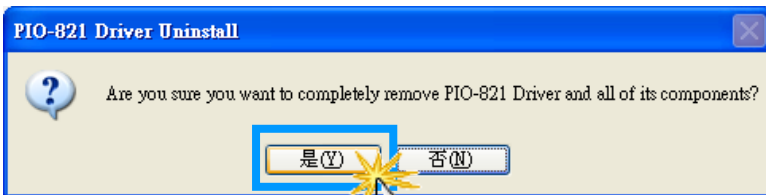
Step 4: Windows pops up **“Found New Hardware”** dialog box again.



## 1.4 Uninstalling the PIO-821 Series Classic Driver

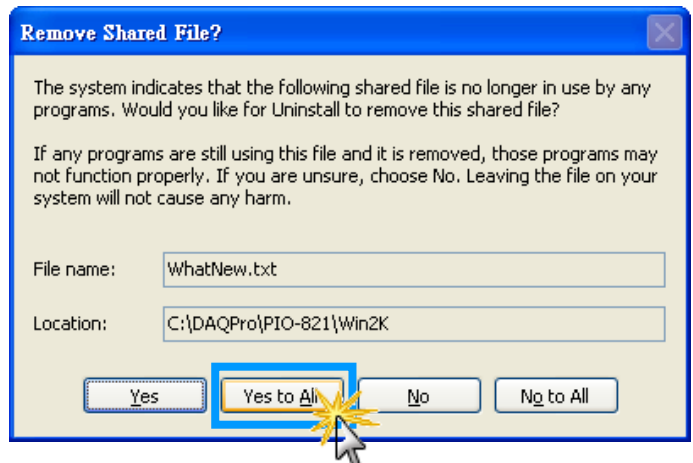
The ICP DAS PIO-821 series classic driver includes an uninstallation utility that allows you remove the software from your computer. To uninstall the software, follow the procedure described below:

Step 1: Double click the **unins000.exe** uninstaller application, which can be found in the following folder:  
**C:\DAQPro\PIO-821.**



Step 2: A dialog box will be displayed asking you to confirm that you want to remove the utility program. Click the “**Yes**” button to continue.

Step 3: The “**Remove Shared File?**” dialog box will then be displayed to confirm whether you want to remove the share files. Click the “**Yes to All**” button to continue.



Step 4: After the uninstallation process is complete, a dialog box will be displayed to you that the driver was successfully removed. Click the “**OK**” button to finish the uninstallation process.

## 2. DLL Function Descriptions

All of the functions provided for PIO-821 series card are listed below in Tables 2-1 to 2-7. This list of functions is expanded on in the text that follows. However, in order to make a clear and simplified description of the functions, the attributes of the input and output parameters for every function is indicated as [input] and [output] respectively, as shown in following table. Furthermore, the error code of all functions supported by PIO-821 is also listed in Section 2-1.

Keyword	Parameter must be set by the user <b>before</b> calling the function	Data/value from this parameter is retrieved <b>after</b> calling the function
[Input]	<b>Yes</b>	<b>No</b>
[Output]	<b>No</b>	<b>Yes</b>

**Table2-1:** Driver Functions Table of PIO821.DLL

Section	Function Definition
<b>2.2</b>	<b>Driver Functions</b>
	WORD PIO821_GetDllVersion();
	WORD PIO821_ActiveBoard(BYTE BoardNo);
	WORD PIO821_CloseBoard(BYTE BoardNo);
	WORD PIO821_TotalBoard();
	WORD PIO821_GetCardInf(BYTE BoardNo, DWORD ID[]);
	BYTE PIO821_IsBoardActive(BYTE BoardNo);

**Table2-2:** D/A Functions Table of PIO821.DLL

Section	Function Definition
<b>2.3</b>	<b>Analog Output Functions</b>
	WORD PIO821_DA_Hex(BYTE BoardNo, WORD wValue);
	WORD PIO821_DA(BYTE BoardNo, BYTE Mode, float fValue);

**Table2-3:** EEPROM Functions Table of PIO821.DLL

Section	Function Definition
<b>2.4</b>	<b>EEPROM Functions</b>
	WORD PIO821_ReadEEP(BYTE BoardNo, WORD *wValue);
	WORD PIO821_WriteEEP(BYTE BoardNo, WORD *wValue);

**Table2-4:** DIO Functions Table of PIO821.DLL

Section	Function Definition
<b>2.5</b>	<b>Digital Input/Output Functions</b>
	WORD PIO821_DigitalIn(BYTE BoardNo, WORD *wValue);
	WORD PIO821_DigitalOut(BYTE BoardNo, WORD wValue);
	BYTE PIO821_InputByte(BYTE BoardNo, DWORD dwOffset);
	void PIO821_OutputByte(BYTE BoardNo, DWORD dwOffset, BYTE bValue);
	WORD PIO821_InputWord(BYTE BoardNo, DWORD dwOffset);
	void PIO821_OutputWord(BYTE BoardNo, DWORD dwOffset, WORD wValue);

**Table2-5:** Timer/Counter Functions Table of PIO821.DLL

Section	Function Definition
<b>2.6</b>	<b>Timer/Counter Functions</b>
	WORD PIO821_SetCounter( BYTE BoardNo, WORD wCounterNo, WORD bCounterMode, DWORD wCounterValue);
	DWORD PIO821_ReadCounter(BYTE BoardNo, WORD wCounterNo, WORD bCounterMode);

**Table2-6:** A/D Functions Table of PIO821.DLL

Section	Function Definition
<b>2.7</b>	<b>Analog Input Functions</b>
	WORD PIO821_SetChannelConfig(BYTE BoardNo, WORD wAdChannel, WORD wConfig);
	WORD PIO821_Delay(BYTE BoardNo,WORD wDownCount)
	WORD PIO821_ADPollingHex(BYTE BoardNo, WORD *wAdVal);
	WORD PIO821_ADPolling(BYTE BoardNo, float *fAdVal);
	WORD PIO821_ADsPolling (BYTE BoardNo, float fAdVal[], DWORD dwNum);
	WORD PIO821_ADsPacer(BYTE BoardNo, float fAdVal[], DWORD dwNum, WORD wSamplingDiv);

**Table2-7:** Interrupt Functions Table of PIO821.DLL

Section	Function Definition
<b>2.8</b>	<b>Interrupt Functions</b>
	WORD PIO821_InstallIrq(BYTE BoardNo);
	WORD PIO821_IntADStart(BYTE BoardNo, WORD wNum, WORD wSamplingDiv);
	WORD PIO821_GetADsfloat(float *fAdVal);
	WORD PIO821_GetADsHex(WORD *HAdVal);
	void PIO821_RemoveIrq(BYTE BoardNo);

## 2.1 Error Code Table

For the most errors, it is recommended to check:

1. Does the device driver installs successful?
2. Does the card have plugged?
3. Does the card conflicts with other device?
4. Close other applications to free the system resources.
5. Try to use another slot to plug the card.
6. Restart your system to try again.

Error Code	Error ID	Error String
0	PIO821_NoError	OK
1	PIO821_ActiveBoardError	This board cannot be activated.
2	PIO821_ExceedFindBoards	The board number exceeds the maximum board number (7).
3	PIO821_DriverNoOpen	Base address is over range.
4	PIO821_BoardNoActive	Base address overlap.
5	PIO821_WriteEEPROMError	Write the EEPROM error
6	PIO821_ModeDAError	DA mode is error
7	PIO821_DAEError	Parameter is null or out of range
8	PIO821_ConfigError	AD gain value is error
9	PIO821_TimeoutError	Delay time out
10	PIO821_AdChannelError	AD channel value is out of range
11	PIO821_AdPollingTimeOut	AD polling is time out
12	PIO821_AdPacerTimeOut	AD pacer is time out
13	PIO821_CounterModeError	Counter value is out of range
14	PIO821_InterruptError	Interrupt is not enable

## 2.2 Driver Functions

### PIO821\_GetDllVersion

Obtain the version information of PIO821.DLL driver.

- **Syntax:**  
WORD **PIO821\_GetDllVersion**(void);
- **Parameters:**  
None
- **Returns:**  
DLL version information.  
For example: If 101(hex) value is return, it means driver version is 1.01.

### PIO821\_ActiveBoard

Activate the device. It must be called once before using the other functions of PIO-821 series boards.

- **Syntax:**  
WORD **PIO821\_ActiveBoard**(BYTE **BoardNo**);
- **Parameters:**  
  
BoardNo  
[Input] Board number 0 to 15 of PIO-821 series.
- **Returns:**  

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_ActiveBoardError	This board can not be activated



## PIO821\_CloseBoard

Stop and close the PIO-821 kernel driver and release the resources of the device from system. This method must be called once before exiting the user's application program.

➤ **Syntax:**

WORD **PIO821\_CloseBoard**(BYTE **BoardNo**);

➤ **Parameters:**

*BoardNo*

[Input] Board number 0 to 15 of PIO-821 series.

➤ **Returns:**

PIO821_NoError	OK
PIO821_BoardNoOpen	The board is not activated
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)

## PIO821\_TotalBoard

Obtain the total board number of PIO-821 series boards installed in the PCI bus.

➤ **Syntax:**

WORD **PIO821\_TotalBoard**(void);

➤ **Parameters:**

None

➤ **Returns:**

Return the total board number.

## PIO821\_GetCardInf

Obtain the information of PIO-821 series boards, which include vender ID, device ID and interrupt number.

➤ **Syntax:**

```
WORD PIO821_GetCardInf(BYTE BoardNo, DWORD ID[]);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

ID[]

[Output] ID[0] → vendor ID of this board

[Output] ID[1] → device ID of this board

[Output] ID[2] → sub-vendor ID of this board

[Output] ID[3] → sub-device ID of this board

[Output] ID[4] → sub-auxiliary ID of this board

[Output] ID[5] → logical interrupt number of this board

➤ **Returns:**

PIO821\_NoError

OK

PIO821\_DriverNoOpen

Kernel driver can not be found

PIO821\_ExceedFindBoards

BoardNo exceeds the current total board number (N)

## PIO821\_IsBoardActive

Obtain the information about the specific board is active or not.

➤ **Syntax:**

```
BYTE PIO821_IsBoardActive(BYTE BoardNo);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

➤ **Returns:**

0 → means the board is inactive.

1 → means the board is active.

## 2.3 Analog Output Functions

### PIO821\_DA\_Hex

Output a 12-bit HEX value to analog output channel.

➤ **Syntax:**

```
WORD PIO821_DA_Hex(BYTE BoardNo, WORD wValue);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

wValue

[Input] Analog output value 0 to 0xfff.

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_BoardNoActive	The board is not activated
PIO821_ParameterError	wValue is out of range

## PIO821\_DA

Output a float value to analog output channel.

➤ **Syntax:**

WORD **PIO821\_DA**(BYTE **BoardNo**, BYTE **Mode**, float **fValue**)

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

Mode

[Input] D/A channel mode 1 or mode2. (Mode1 → 5 V, Mode2 → 10 V)

fValue

[Input] Analog output value.

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_BoardNoActive	The board is not activated
PIO821_ParameterError	wValue is out of range

## 2.4 EEPROM Functions

### PIO821\_WriteEEP

Write 64 words (128 bytes) data into the EEPROM of the PIO-821 series board. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

```
WORD PIO821_WriteEEP(BYTE BoardNo, WORD *wValue);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

\*wValue

[Input] Read first WORD (16-bit) of data.

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_BoardNoActive	The board is not activated
PIO821_WriteEEPROMError	Fail to write data to EEPROM

## 2.5 Digital Input/Output Functions

### PIO821\_DigitalIn

Obtain the 16 TTL-compatible digital input values from the PIO-821 series board. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

```
WORD PIO821_DigitalIn(BYTE BoardNo, WORD *wValue);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

\*wValue

[Output] Read the digital input value.

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)

## PIO821\_DigitalOut

Send out digital value through 16 TTL-compatible digital output channels. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

WORD PIO821\_DigitalOut(BYTE BoardNo, WORD wValue);

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

wValue

[Input] Digital output value.

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)



## PIO821\_InputByte

Obtain a byte data from the specific address mapping of the PIO-821 series board. Please call PIO821\_ActiveBoard first before using this function. This function is designed for advance user to access the hardware data based on the register of PIO-821 series.

➤ **Syntax:**

```
BYTE PIO821_InputByte(BYTE BoardNo, DWORD dwOffset);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

dwOffset

[Input] The offset value of the base address of the PIO-821 series board for the mapping address, from 0 to 0xff.

➤ **Returns:**

One Byte value or data.

## PIO821\_OutputByte

Write a byte data to the defined address of the PIO-821 series board. This function is designed for advance user to write data into the hardware based on the register of PIO821 series.

➤ **Syntax:**

```
void PIO821_OutputByte(BYTE BoardNo, DWORD dwOffset, BYTE bValue);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

dwOffset

[Input] The offset value of the base address of the PIO-821 series board for the mapping address, from 0 to 0xff.

bValue

[Output] A Byte value for output.

➤ **Returns:**

None

## PIO821\_InputWord

Obtain a word (two bytes) data from the specific mapping address of the PIO-821 series board. Please call PIO821\_ActiveBoard first before using this function. This function is designed for advance users to access the hardware data based on the register of PIO-821 series.

➤ **Syntax:**

```
WORD PIO821_InputWord(BYTE BoardNo, DWORD dwOffset);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

dwOffset

[Input] The offset value of the base address of the PIO-821 series board for the mapping address, from 0 to 0xff.

➤ **Returns:**

One WORD value or data.

## PIO821\_OutputWord

Write a word( two bytes) data to the defined address of the PIO-821 series board. This function is designed for advance user to write into the hardware based on the register of PIO-821 series.

➤ **Syntax:**

```
void PIO821_OutputWord(BYTE BoardNo, DWORD dwOffset, WORD wValue);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

dwOffset

[Input] The offset value of the base address of the PIO-821 series board for the mapping address, from 0 to 0xff.

wValue

[Output] A WORD value for output.

➤ **Returns:**

None

## 2.6 Timer/Counter Functions

### PIO821\_SetCounter

Set the counter number, configuration code and counter value to the 8254 chip of PIO-821 series board. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

```
WORD PIO821_SetCounter( BYTE BoardNo, WORD wCounterNo, WORD bCounterMode,  
                        DWORD wCounterValue);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

wCounterNo

[Input] Select the 8254 Counter0 to Counter2.

bCounterMode

[Input] The configuration code. Please refer to specification of 8254 chip.

wCounterValue

[Input] Counter value of 8254 chip.

➤ **Returns:**

PIO821_NoError	OK
PIO821_CounterModeError	Out of counter mode range

## PIO821\_ReadCounter

Read the counter value from the specified counter. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

```
DWORD PIO821_ReadCounter(BYTE BoardNo, WORD wCounterNo, WORD bCounterMode);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

wCounterNo

[Input] Select the 8254 Counter0 to Counter2.

bCounterMode

[Input] The configuration code. Please refer to specification of 8254 chip.

➤ **Returns:**

PIO821_NoError	OK
PIO821_CounterModeError	Out of counter mode range

## 2.7 Analog Input Functions

### PIO821\_SetChannelConfig

Set the channel configuration for analog input, which includes AD channel number and Gain mode. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

WORD PIO821\_SetChannelConfig(BYTE BoardNo, WORD wAdChannel, WORD wConfig);

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

wAdChannel

[Input] Select A/D channel number 0 to 16.

wConfig

[Input] Select A/D channel gain, refer to section 7.3.12 “A/D Gain Control and Multiplex Control Register” of the PIO-821 hardware manual.

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_AdChannelError	Out of the number value of channel
PIO821_ConfigError	Out of the gain value of channel

## PIO821\_Delay

Use the 8254 chip to delay the specific time waiting in the program.

➤ **Syntax:**

WORD **PIO821\_Delay**(BYTE **BoardNo**, WORD **wDownCount**)

➤ **Parameters:**

*BoardNo*

[Input] Board number 0 to 15 of PIO-821 series.

*wDownCount*

[Input] Counter's value of 8254 chip.

➤ **Returns:**

PIO821_NoError	OK
PIO821_TimeoutError	Out of the delay time



## PIO821\_ADPollingHex

Read a 12-bit HEX value from the specified analog input channel. The active AD is setting by PIO821\_SetChannelConfig(...).This subroutine performs the AD conversion by polling one time. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

WORD PIO821\_ADPollingHex(BYTE BoardNo, WORD \*wAdVal);

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

\*wAdVal

[Output] Address of wAdVal which store the AD HEX data (12 bits).

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_AdPollingTimeOut	AD polling is time out

## PIO821\_ADPolling

Read a the value of current active AD from the analog input channel. The active AD is set by PIO821\_SetChannelConfig(...). This subroutine performs the AD conversion by polling one time. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

```
WORD PIO821_ADPolling(BYTE BoardNo, float *fAdVal);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

\*wAdVal

[Output] Address of wAdVal which store the AD data (12 bits).

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_BoardNoActive	The board is not activated
PIO821_AdPollingTimeOut	AD polling is time out

## PIO821\_ADsPolling

Read multiple the values of current active AD from the analog input channel. The active AD channel is set by PIO821\_SetChannelConfig(...). This subroutine performs the AD conversions by polling trigger. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

WORD PIO821\_ADsPolling (BYTE BoardNo, float fAdVal[], DWORD dwNum);

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

fAdVal[]

[Output] Piece address of fAdVal which store the A/D data (12 bits).

dwNum

[Input] Number of A/D conversions will be performed.

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_BoardNoActive	The board is not activated
PIO821_AdPollingTimeOut	AD polling is time out

## PIO821\_ADsPacer

Read multiple the values of current active AD from the analog input channel. The active AD channel is set by PIO821\_SetChannelConfig(...). This subroutine performs the AD conversions by pacer trigger. Please call PIO821\_ActiveBoard first before using this function.

➤ **Syntax:**

```
WORD PIO821_ADsPacer(BYTE BoardNo, float fAdVal[], DWORD dwNum, WORD  
                    wSamplingDiv);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

fAdVal[]

[Output] Piece address of fAdVal which store the A/D data (12 bits).

dwNum

[Input] Number of A/D conversions will be performed.

wSamplingDiv

[Input] A/D sampling rate = 2 M/wSamplingDiv

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)
PIO821_BoardNoActive	The board is not activated
PIO821_AdPacerTimeOut	AD pacer is time out

## 2.8 Interrupt Functions

### PIO821\_InstallIrq

This function can enable the interrupt service for the specific PIO821 card. After applying the function, the system would allocate a handle to the interrupt.

➤ **Syntax:**

```
WORD PIO821_InstallIrq(BYTE BoardNo);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

➤ **Returns:**

PIO821_NoError	OK
PIO821_InterruptError	Interrupt enable is error

### PIO821\_IntADStart

This function uses the interrupt method to read and store the AD values. Users must apply the PIO821\_SetChannelConfig function to configure the specific AD channel first.

➤ **Syntax:**

```
WORD PIO821_IntADStart(BYTE BoardNo, WORD wNum, WORD wSamplingDiv);
```

➤ **Parameters:**

BoardNo

[Input] Board number 0 to 15 of PIO-821 series.

wNum

[Output] Number of interrupt A/D conversions will be performed.

wSamplingDiv

[Input] A/D sampling rate = 2 M/wSamplingDiv

➤ **Returns:**

PIO821_NoError	OK
PIO821_DriverNoOpen	Kernel driver can not be found
PIO821_ExceedFindBoards	BoardNo exceeds the current total board number (N)

## PIO821\_GetADsfloat

The function can get the float AD data of the specific AD channel. Users can set the specific AD channel in PIO821\_SetChannelConfig function. And the data is from the interrupt method after applying PIO821\_IntADStart function.

➤ **Syntax:**

WORD PIO821\_GetADsfloat(float \*fAdVal);

➤ **Parameters:**

\*fAdVal

[Output] Start address of fAdVal which store the A/D data.

➤ **Returns:**

Interrupt statue: (0) data is incomplete  
(1) data is complete

## PIO821\_GetADsHex

The function can get the hex-format AD data of the specific AD channel. Users can set the specific AD channel in PIO821\_SetChannelConfig function. And the data is from the interrupt method after applying PIO821\_IntADStart function.

➤ **Syntax:**

```
WORD PIO821_GetADsHex(WORD *HAdVal);
```

➤ **Parameters:**

*\*HAdVal*

[Output] Start address of HAdVal which store the A/D data (12 bits).

➤ **Returns:**

Interrupt statue: (0) data is incomplete  
(1) data is complete

## PIO821\_RemoveIrq

Release the interrupt resource of specific board from the computer system.

➤ **Syntax:**

```
void PIO821_RemoveIrq(BYTE BoardNo);
```

➤ **Parameters:**

*BoardNo*

[Input] Board number 0 to 15 of PIO-821 series.

➤ **Returns:**



Null

## 3. Demo Programs

### 3.1 For Microsoft Windows

ICP DAS PIO-821 Series Classic Driver DLL contains a set of functions. It can be used in various application programs for PIO-821 series card. The API functions supports many development environments and programming languages, including Microsoft Visual C++ , Visual Basic , Borland Delphi , Borland C Builder++ , Microsoft Visual C#.NET , Microsoft Visual VB.NET.

The demo programs of Windows OS for the PIO-821 series can be found on the supplied CD-ROM, or can be obtained from the ICP DAS FTP web site. The location and addresses are indicated in the table below:

	CD:\NAPDOS\PCI\PIO-821\DLL\Demo\ 
	<a href="http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-821/dll/demo/">http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-821/dll/demo/</a>

<ul style="list-style-type: none"><li>⊕ BCB4 → for Borland C++ Builder 4</li><li>PIO821.H → Header files</li><li>PIO821.LIB → Linkage library for BCB only</li></ul>	<ul style="list-style-type: none"><li>⊕ Delphi4 → for Delphi 4</li><li>PIO821.PAS → Declaration files</li></ul>
<ul style="list-style-type: none"><li>⊕ VC6 → for Visual C++ 6</li><li>PIO821.H → Header files</li><li>PIO821.LIB → Linkage library for VC only</li></ul>	<ul style="list-style-type: none"><li>⊕ VB6 → for Visual Basic 6</li><li>PIO821.BAS → Declaration files</li></ul>
<ul style="list-style-type: none"><li>⊕ VB.NET2005 → for VB.NET2005</li><li>PIO821.vb → Visual Basic Source files</li></ul>	<ul style="list-style-type: none"><li>⊕ CSharp2005 → for C#.NET2005</li><li>PIO821.cs → Visual C# Source files</li></ul>

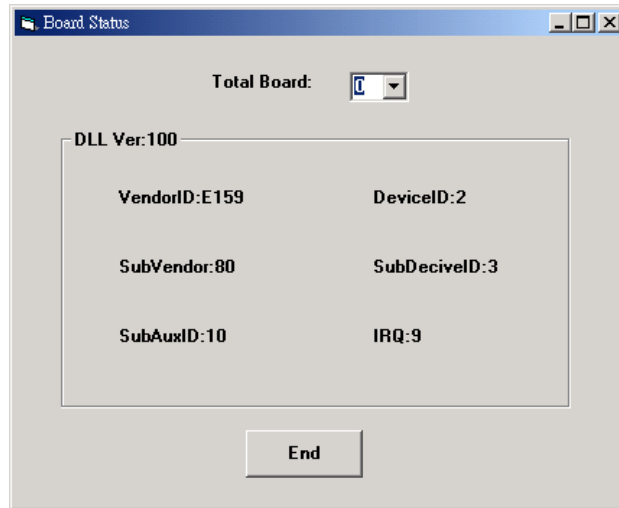
#### The list of demo programs:

- ⊕ Config Demo: Get cards information
- ⊕ Counter Demo: Counter demo
- ⊕ DIO Demo: Digital Input and digital output
- ⊕ Interrupt Demo: Get the AD value by interrupt method
- ⊕ Pacer Demo: Get the AD value by pacer method
- ⊕ Polling Demo: Get the AD value by polling method



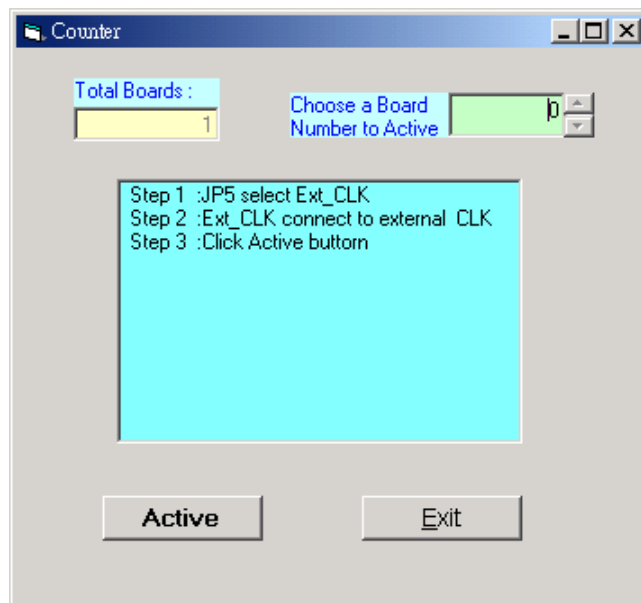
## Config Demo: Get cards information

Following figure is the result for the demo1 program. It can be applied to obtain the hardware information of the PIO-821 board.



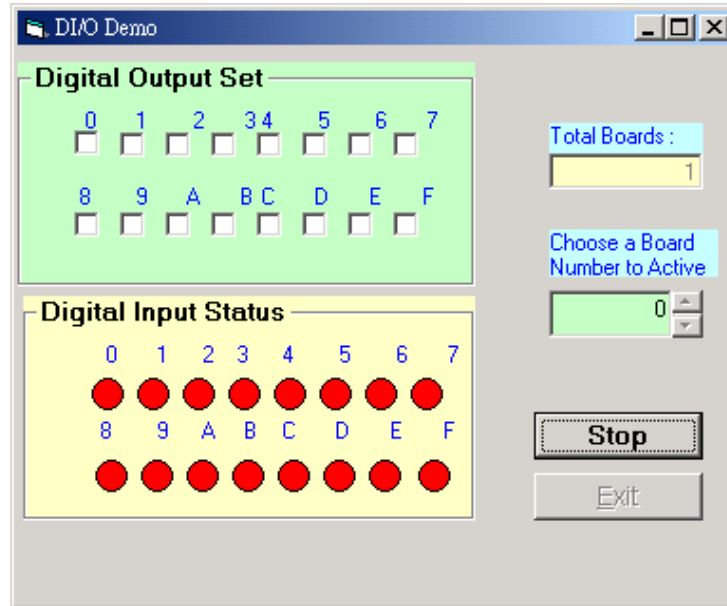
## Counter Demo: Counter demo

This demo program can be used to obtain the counter0 information of 8254 chip on board. And users can set the external clock of the hardware by setting JP5 jumper. Click the "Active" button to show the count value of the external signal.



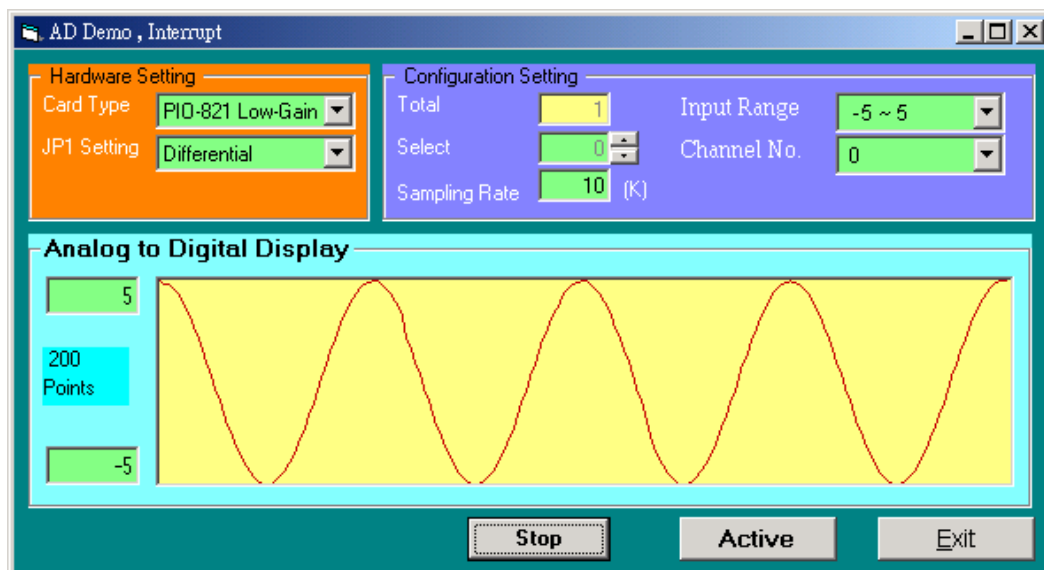
## DIO Demo: Digital input/output

This program demonstrates the DI/DO status of PIO-821 board after the digital input/output wire connection.



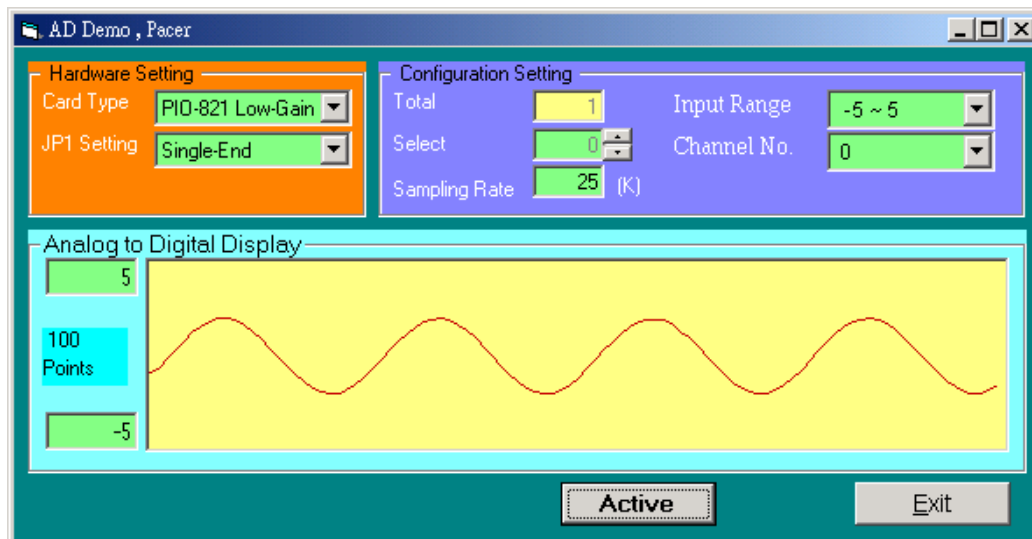
## Interrupt Demo: The interrupt method to get the AD value

This demo program shows the AD value by the interrupt method. Users can set the Input range and sampling rate of AD channel in this demo and click "show" button to get the analog input value and demonstrate the data in the display window.



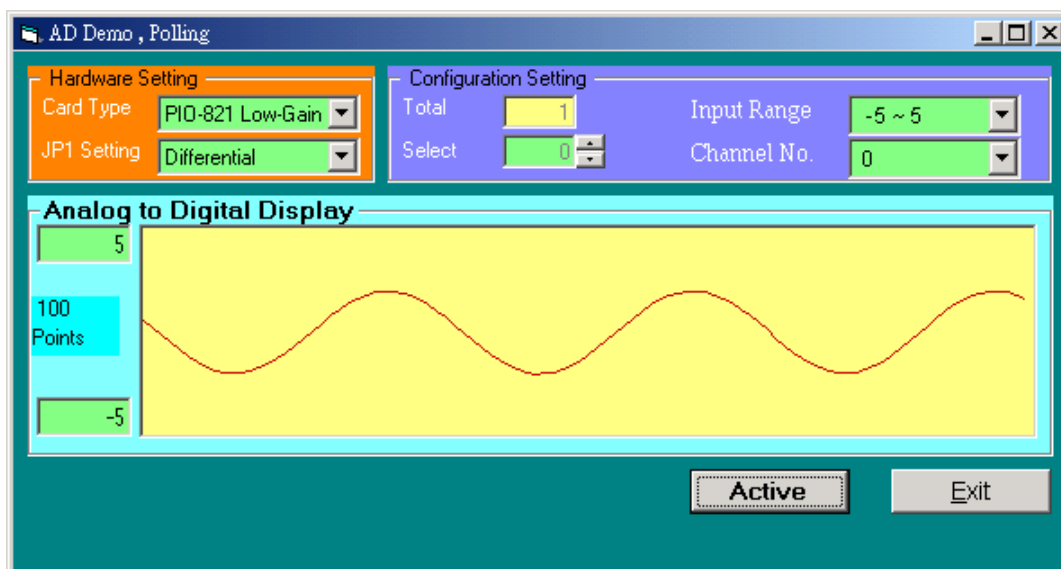
## Pacer Demo: The pacer mode to get the AD value

This demo program provides the pacer method to get the AD value.



## Polling Demo6: The Polling mode to get the AD value

This demo program provides the polling method to get the AD value.



## 3.2 For DOS

The demo program is contained in:



CD:\NAPDOS\PCI\PIO-821\DOS\



<http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-821/dos/>

The completely source listing of demo program is given in TC format. This program is compiler in LARGE mode and link with PIO.lib in TC.

⊕ \TC\\*. \* → for Turbo C 2.xx or above

⊕ \TC\LIB\\*. \* → for TC Library

⊕ \TC\DEMO\\*. \* → for TC demo program

⊕ \TC\DIAG\\*. \* → for TC diagnostic program

⊕ \TC\LIB\PIO.H → TC Declaration File

⊕ \TC\LIB\TCPIO\_L.LIB → TC Large Model Library File

⊕ \TC\LIB\TCPIO\_H.LIB → TC Huge Model Library File

### The list of demo programs:

⊕ DIO: DIO Test

⊕ DA: Analog output test

⊕ Wave: 8254 square wave generator

⊕ EEPROM: Save EEPROM data to file

⊕ Cal: Digital to Analog output without calibration

⊕ Softtrg: Analog to Digital by Software trigger without calibration

⊕ Pacerca: Analog to Digital by Pacer trigger without calibration

⊕ Pacer: Analog to Digital by Pacer trigger with calibration

***Note that all of the hardware control functions need to be provided and processed by user themselves.***

## 3.2.1 LIB (PIO.H) Function Description

---

### PIO\_FloatSub2

---

Compute  $C=nA-nB$  in **float** format, which is 32 bits floating pointer number. This function is provided for testing purpose.

- **Syntax:**  
float **PIO\_FloatSub2** (float **fA**, float **fB**);
- **Parameters:**  
  
*fA*: float point value  
  
*fB*: float point value
- **Returns:**  
Return float point value ( $fA - fB$ )

### PIO\_ShortSub2

---

Compute  $C=nA-nB$  in **short** format, short=16 bits sign integer. This function is provided for testing purpose.

- **Syntax:**  
float **PIO\_ShortSub2** (short **nA**, short **nB**);
- **Parameters:**  
  
*nA*: Short integer  
  
*nB*: Short integer
- **Returns:**  
Return a short integer ( $nA - nB$ )

---

## PIO\_GetDriverVersion

---

Obtain the software version

➤ **Syntax:**

WORD **PIO\_GetDriverVersion** (WORD **\*wDriverVersion**);

➤ **Parameters:**

*\*wDriverVersion*: Driver Version.

For example: If 101(hex) is return, it means driver version is 1.01

➤ **Returns:**

NoError

---

## PIO\_DriverInit

---

This function searches the hardware board. If all checks are OK, this function will return the total board value.

➤ **Syntax:**

WORD **PIO\_DriverInit**(WORD **\*wBoards**, WORD **wSubVendorID**, WORD **wSubDeviceID** ,WORD **wSubAuxID**);

➤ **Parameters:**

*\*wBoards*: **[Output]** Total board

*wSubVendorID*: **[Input]** Sub Vendor ID of PIO-821 series card

*wSubDeviceID*: **[Input]** Sub Device ID of PIO-821 series card

*wSubAuxID*: **[Input]** Axu ID of PIO-821 series card

➤ **Returns:**

Null

---

## PIO\_GetConfigAddressSpace

---

Get configuration address space of PIO-821 series card.

➤ **Syntax:**

```
WORD PIO_GetConfigAddressSpace(WORD wBoardNo, WORD *wBaseAddr, WORD *wIrq,  
                                WORD *wSubVendor, WORD *wSubDevice, WORD  
                                *wSubAux, WORD *wSlotBus, WORD *wSlotDevice)
```

➤ **Parameters:**

wBoardNo: [Input] board number(0 to 7)

\*wBaseAddr: [Output] Base address

\*wIrq: [Output] IRQ number

\*wSubVendor: [Output] Sub Vendor ID

\*wSubDevice: [Output] Sub Device ID

\*wSubAux: [Output] Sub Aux ID

\*wSlotBus: [Output] PCI slot

\*wSlotDevice: [Output] Device of slot

➤ **Returns:**

NoError: OK.

FindBoardError: Cannot find the PIO-821 series card.